

Automatic Separation of Compound Figures in Scientific Articles

Mario Taschwer · Oge Marques

the date of receipt and acceptance should be inserted later

Abstract

Content-based analysis and retrieval of digital images found in scientific articles is often hindered by images consisting of multiple subfigures (compound figures). We address this problem by proposing a method to automatically classify and separate compound figures, which consists of two main steps: (i) a supervised compound figure classifier (CFC) discriminates between compound and non-compound figures using task-specific image features; and (ii) an image processing algorithm is applied to predicted compound images to perform compound figure separation (CFS). Our CFC approach is shown to achieve state-of-the-art classification performance on a published dataset. Our CFS algorithm shows superior separation accuracy on two different datasets compared to other known automatic approaches. Finally, we propose a method to evaluate the effectiveness of the CFC-CFS process chain and use it to optimize the misclassification loss of CFC for maximal effectiveness in the process chain.

Keywords multipanel figure separation · document image understanding

1 Introduction

The work described in this paper is motivated by the realization that articles in scientific publications contain a substantial amount of figures consisting of two or more subfigures, which could be treated as separate images for the purpose

Mario Taschwer
ITEC, Klagenfurt University (AAU), Austria
E-mail: mario.taschwer@aau.at

Oge Marques
Florida Atlantic University (FAU), Boca Raton, FL, USA
E-mail: omarques@fau.edu

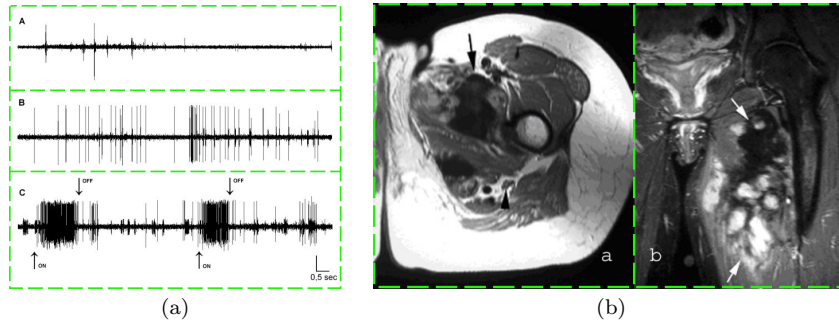


Fig. 1 Sample compound images (of the ImageCLEF 2015 CFS dataset [11]) suitable for two different separator detection algorithms. Subfigures are separated by (a) whitespace, (b) a vertical edge. Dashed lines represent the expected output of CFS.

of automatic content-based analysis or indexing for retrieval. Figure 1 shows two examples of such *compound figures* found in a dataset of article images of the biomedical literature. Based on published datasets drawn from open access biomedical literature, it has been estimated that between 40% and 60% of figures occurring in articles are compound figures [1,8,10].

In this paper we address the problem of automatically recognizing and separating compound figures in a collection of article images by breaking it into two subproblems: *compound figure classification* (CFC) and *compound figure separation* (CFS). CFC is a binary classification problem that aims at discriminating between compound and non-compound figures given an article image. CFS is the problem of determining the bounding boxes of all subfigures of a given compound figure. Algorithms solving the CFC and CFS problems are naturally combined into a *CFC-CFS process chain* that receives arbitrary article images as input and delivers bounding boxes of subfigures (or of single figures) at the output. Images classified as *compound* by the CFC algorithm are further processed by CFS, whereas for images predicted as *non-compound* by the CFC a bounding box covering the entire image is produced.

For CFC, we propose several global image features designed to capture the existence of edges or whitespace that could potentially separate subfigures and use them with well-known supervised machine learning algorithms. For CFS, we designed an image processing algorithm comprising distinct modules for detecting two types of separators between subfigures: (1) homogeneous rectangular areas of whitespace spanning the entire image width or height, which we call *separator bands* (shown in Fig. 1(a)); and (2) *separator edges* spanning the entire image width or height, which may arise from borders drawn around subfigures or from adjacent subfigures “stitched together” as shown in Fig. 1(b). The proposed CFS algorithm internally uses a separate binary classifier (independent from CFC) to decide which of the two separator detection modules to apply to a given compound image. Based on the observation that compound images containing graphical illustrations (such as diagrams and charts) often contain separator bands, whereas most subfigures in other

compound images show rectangular border edges, we train the internal CFS classifier to discriminate between graphical illustrations and other article images and call it *illustration classifier*.

This paper is based on previous work [26, 27] and provides the following additional research contributions:

1. It proposes novel image features for compound figure classification, which can be efficiently extracted and achieve state-of-the-art CFC performance using well-known classifier algorithms.
2. It demonstrates that the proposed CFS algorithm outperforms state-of-the-art automatic and semi-automatic CFS approaches on two recently published biomedical datasets.
3. It establishes a method to evaluate CFC-CFS chain effectiveness, which is applied to investigate the effect of various CFC implementations in the chain.

1.1 Motivation and Context

From a problem-oriented point of view, research on automatic compound figure separation is motivated by the fact that compound figures hamper content-based analysis and indexing of article images for retrieval, because global image features extracted from a compound image are a mixture (often an average) of the same features extracted from the subfigures only, leading to reduced discriminative power of these features on compound images. The situation may be slightly better for local image features, which capture the existence of certain texture or shape patterns in small image regions, but the predominant way of aggregating local features of an image in a Bag of Visual Words representation [23] still suffers from the additive effect of including local features from all subfigures. Moreover, subfigures of a given compound image usually convey different semantic information that may be relevant for retrieval, although the compound figure establishes a common semantic context for subfigures.

From a historical perspective, the research community showed little interest in the CFC and CFS problems until ImageCLEF 2013, where a CFS task was introduced as one of the challenges in the biomedical domain [10]. Task organizers provided training and test datasets, and evaluated CFS results submitted by participants for the test dataset. It is presumably this provisioning of datasets and of a CFS evaluation method that stimulated research on CFC and CFS problems in recent years. Since we are not aware of any CFC and CFS datasets available to the research community other than the ones described in Section 3.1, our experiments are limited to the biomedical domain, although our proposed algorithms should be applicable to other scientific domains as well.

1.2 Related Work

There is little work about the CFC problem in the literature, but research interest may grow due to a CFC task introduced recently at ImageCLEF 2015 [11]. From the two participating groups of this task, Pelka and Friedrich [18] achieved best results (an accuracy of 85.4%) using both textual and visual features with a random forest classifier. Textual features were extracted from image captions, and visual features were derived from detected separator bands and a bag-of-keypoints representation of dense-sampled SIFT keypoints. Their submitted variant using visual features only resulted in 72.5% accuracy. Wang et al. [28], the other participating group in the ImageCLEF 2015 CFC task, achieved 82.8% accuracy using visual features only. They used an unsupervised approach consisting of connected component analysis followed by separator band detection. A separate evaluation run using connected component analysis only resulted in 82.5% accuracy, so separator band detection had an almost negligible effect in the combined approach.

Prior to ImageCLEF 2015, Yuan and Ang [29] proposed a 3-class classifier discriminating between photographs, non-photographs, and compound images containing both photographs and non-photographs. The classifier was used in a process chain for CFS, but it is not clear whether the classifier output should affect CFS operation. Experiments did not include classifier evaluation.

An established research field whose techniques could potentially be useful for CFC is document image classification [7]. Although it deals with digital images of entire document pages (containing mostly text), some of the proposed methods – such as block segmentation and physical layout analysis – may also help CFC. However, there is no evidence of such utilization in the literature yet.

The success of deep learning techniques [2] or other advanced methods of representation learning [3] in image classification tasks during the last decade [15, 25] suggest that they could also be applied to the CFC problem. However, we believe that available CFC training sets are still too small to obtain effective classifiers from deep learning methods, and we hope that recently proposed “simple” CFC methods (including ours) will help build larger training sets for advanced machine learning techniques.

Regarding the CFS problem, most existing approaches focus on the detection of separator bands [1, 8, 13, 29] and hence fail for compound images where subimages are stitched together without separator bands (see Fig. 1(b)). Apostolova et al. [1] propose to solve the CFS problem using not only visual information contained in article images, but also textual information contained in images (extracted using OCR techniques) and in image captions. Since our proposed CFS algorithm does not use textual information, we compare it to their visual CFS algorithm (described as *image panel segmentation*), which is part of a five-stage process chain and includes image markup removal. Image markup consists of text labels embedded in compound images that may be located in separator bands, exacerbating separator detection. Their CFS

method [1] has recently been used in an approach to document image classification and retrieval by Simpson et al. [22].

Chhatkuli et al. [8] employ several image preprocessing techniques – including binarization, border cropping, and image markup removal – prior to detecting separator bands. Separator band detection is done recursively in horizontal direction first, followed by recursive detection of vertical separators, which may lead to limited separation of irregular subfigure structures. Separator candidates are filtered by a complex rule-based analysis step. Evaluation is performed using a self-constructed dataset and evaluation method based on separator locations, making a comparison with our approach infeasible.

Kitanovski et al. [13] participated in the ImageCLEF 2013 CFS task using an apparently simple approach based on separator band detection. They do not provide details and achieved an accuracy of 69%.

Yuan and Ang [29] build upon the approach of Murphy et al. [17] and of Qian and Murphy [19] and use a sliding window to compute intensity histograms of horizontal and vertical bands to detect (white) separator bands. Additionally, they used an edge-based approach involving Hough transform to separate overlaid zoom-in views from background figures, a case that is not considered in this work. They evaluated their CFS method on two self-constructed small datasets of about 180 compound figures each, but did not provide enough details to make their evaluation method reproducible.

A different approach to CFS is based on connected component analysis of binarized images, which, however, is susceptible to over-segmentation, in particular for subfigures containing diagrams or charts. Shatkay et al. [21] used such a technique for CFS in the context of document classification, but did not evaluate CFS effectiveness separately. The CFC approach proposed by Wang et al. [28] determines subfigures using connected component analysis that could probably also be used in a CFS algorithm.

The approach of NLM (U.S. National Library of Medicine) [20] and our previous approach [26], both submitted to the ImageCLEF 2015 [11] CFS task, independently proposed to address compound images without separator bands by processing edge detection results. Besides algorithmic differences in edge-based separator detection, our approach incorporates a classifier to automatically select edge- or band-based separator detection, whereas NLM’s approach uses manual image classification for evaluation.¹

1.3 Structure of the Paper

The proposed methods to address the CFC and CFS problems are described in Section 2, including a way to improve the effectiveness of the CFC-CFS process chain (Section 2.3). Section 3 explains the experimental setup to evaluate our approach and, in particular, describes the datasets (Section 3.1) and evaluation methods (Section 3.2) used. Evaluation results are presented and discussed in

¹ We therefore call NLM’s approach [20] *semi-automatic*, although an automatic classifier could be easily integrated.

the same section, separately for CFC (Section 3.3), CFS (Section 3.4), and the CFC-CFS process chain (Section 3.5). Section 4 concludes the paper and makes some suggestions for future work.

2 Methods

In the following two subsections, we describe our proposed approach to address the CFC and CFS problems, respectively. A technique to improve the effectiveness of the CFC-CFS process chain, given an imperfect CFC implementation, will be described in Section 2.3.

2.1 Compound Figure Classifier

Recognizing compound figures in a dataset of article images can be viewed as a binary classification problem. We address this problem by using hand-crafted image features and classical machine learning algorithms, because we consider the available training datasets as being too small for deep learning techniques (see Section 1.2), and we expect that the effect of limited classification accuracy on the CFC-CFS process chain can be partly compensated by biasing the classifier towards the *compound* class (see Section 2.3).

For CFC, we propose to use three types of image features determined separately for vertical and horizontal directions of a gray-scale image whose pixel values have been normalized to the range $[0, 1]$. Each feature type is computed by aggregating each pixel line in direction D (vertical or horizontal) to a single real number, resulting in a single *projection vector* representing the image along direction D' orthogonal to D . The spatial distribution of values in the projection vector is then captured by a *spatial profile vector* of fixed length. The final feature vector is formed by concatenating the horizontal profile vectors of the three feature types, followed by the corresponding vertical profile vectors.

The three feature types differ in how the projection vector is calculated: (1) *mean* gray values along pixel lines, (2) *variance* of gray values along pixel lines, and (3) one-dimensional *Hough transform*, which counts the number of edge points aligned in direction D in a binary edge map of the input image. The binary edge map is produced by applying a gradient threshold on edges in direction D detected by the Sobel operator. Hough transform values are then normalized to the range $[0, 1]$ using the image dimension in direction D (width or height). Some of the spatial profile methods applied afterwards require *quantization* of projection vectors, which is performed differently for the three feature types, using quantization parameters (positive integers) p , q , and h :

- Mean projection values are quantized into p bins dividing $[0, 1]$ into p subintervals with lower bounds $1 - 2^{i-p}$ for $i = 1, 2, \dots, p$. The logarithmic scale

for quantization should help to discriminate between high values (white separator bands) and others.

- Variance projection values are quantized into q bins dividing $[0, 1]$ into q subintervals with upper bounds 2^{i-q} for $i = 1, 2, \dots, q$. The logarithmic scale for quantization should help to discriminate between low-variance pixel lines (subfigure separators) and others.
- Normalized Hough transform values are quantized into h bins dividing $[0, 1]$ into h subintervals with lower bounds $1 - 2^{i-h}$ for $i = 1, 2, \dots, h$. The logarithmic scale for quantization should help to discriminate between Hough peaks (subfigure separators) and others.

We consider six spatial profile methods to produce profile vectors from projection vectors. Five of them require quantization of projection vectors and divide the vector of dimensionality N into k *spatial bins* of $\lfloor N/k \rfloor$ or $\lfloor N/k \rfloor + 1$ adjacent positions. An additional profile method tries to capture the spatial structure of the projection vector using its Fast Fourier Transform (FFT).

- *Profile 1*: A spatial bin is represented by the full normalized histogram of quantized projection values, resulting in p , q , or h values per spatial bin.
- *Profile 2*: A spatial bin is represented by the quantized projection value that occurs most often (the mode). This value is then normalized to the range $[0, 1]$.
- *Profile 3*: A spatial bin is represented by the relative frequency of the largest quantized projection value, resulting in a single number in the range $[0, 1]$.
- *Profile 4*: A spatial bin is represented by its maximum quantized projection value, normalized to the range $[0, 1]$.
- *Profile 5*: A spatial bin is represented by its average quantized projection value, normalized to the range $[0, 1]$.
- *Profile 6*: The absolute values of the first k low-frequency FFT coefficients of the projection vector are normalized by $1/N$, such that resulting values are constrained to the range $[0, 1]$.

The dimensionality of feature vectors depends on parameters k , p , q , h , and on the profile method used for each of the three feature types, as presented in Table 1. We denote a certain *feature set* by three numbers xyz representing the spatial profile numbers of mean projection (x), variance projection (y), and Hough Transform (z). A value of zero (e.g. $x = 0$) means that the corresponding component of the feature vector has been dropped. For example, the feature set 034 denotes a feature vector formed by concatenation of horizontal profiles of variance projection and Hough Transform, followed by corresponding vertical profiles. Both profile methods (3 and 4) represent a spatial bin by a single number, resulting in k numbers per profile vector, $2k$ numbers for both horizontal profiles, and $4k$ numbers for the final feature vector.

As classifier algorithms we use logistic regression, a linear support vector machine (SVM), and a non-linear SVM with a radial basis function kernel.

Table 1 Dimensionality of various feature sets used for compound figure classification. k denotes the number of spatial bins used to compute profile vectors. p , q , and h are quantization parameters. The right-most column gives the dimensionality for parameter settings $k = 16$, $p = 5$, $q = 8$, $h = 3$.

Feature Set	Dimensionality	Example
111	$2 * k * (p + q + h)$	512
222	$6 * k$	96
333	$6 * k$	96
444	$6 * k$	96
555	$6 * k$	96
666	$6 * k$	96
011	$2 * k * (q + h)$	352
034	$4 * k$	64
134	$2 * k * (p + 2)$	224
434	$6 * k$	96

2.2 Compound Figure Separation

Our approach to compound figure separation is a recursive algorithm (see Fig. 2) which consists of the following steps: (1) classification of the compound image as illustration or non-illustration image, (2) removal of border bands, (3) detection of separator lines, (4) vertical or horizontal separation, and (5) recursive application to each subfigure image. The *illustration classifier* is used to decide which of two separator line detection modules to apply: if the compound image is classified as an illustration image, the *band-based* algorithm is applied, which aims at detecting separator bands between subfigures. Otherwise, the image is processed by the *edge-based* separator detection algorithm, which applies edge detection and Hough transform to locate candidate separator edges. The algorithm selection is based on the assumption that edge-based separator detection is better suited for non-illustration compound images due to visible vertical or horizontal edges separating subfigures. Note that this assumption is not violated by non-illustration compound images with separator bands where subfigures have a visible rectangular border. The following four sections describe the illustration classifier, the main recursive algorithm, and the two separator detection modules in more detail.

2.2.1 Illustration Classifier

The illustration classifier is used to decide which separator detection algorithm to apply to a given compound image. If the image is predicted to be a graphical illustration with probability greater than `decision.threshold`, the band-based separator detection is applied, otherwise the edge-based separator module is used. This decision is made only once for each compound image, so all recursive invocations use the same separator detection algorithm.

Due to promising effectiveness for CFS in early experiments, we use four sets of global image features as classifier input, computed after gray-level conversion: (1) *simple2* is a two-dimensional feature consisting of image entropy,

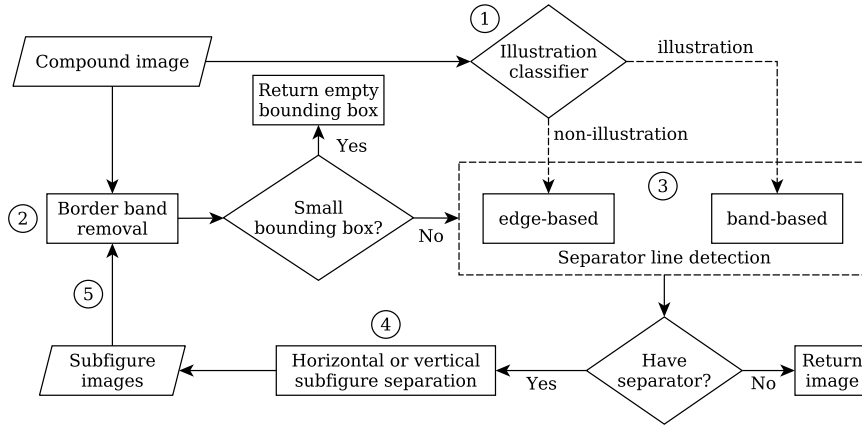


Fig. 2 Recursive algorithm for compound figure separation. Numbers denote the main algorithmic steps described in the beginning of Section 2.2.

estimated using a 256-bin histogram, and mean intensity; (2) *simple11* extends *simple2* by 9 quantiles of the intensity distribution; (3) *CEDD* is the well-known color and edge directivity descriptor [6] (144-dimensional); and (4) *CEDD_simple11* is the concatenation of *CEDD* and *simple11* features (155-dimensional).

As machine learning algorithms we consider support vector machines (SVM) with radial basis function kernel (RBF) and logistic regression. Although logistic regression is generally inferior to kernel SVM due to its linear decision boundary, it has the advantage of providing prediction probabilities, which allow us to tune the selection of separator detection algorithms using the `decision_threshold` parameter.

2.2.2 Recursive Algorithm

Before applying the main algorithm (Fig. 2) to a given compound figure image, it is converted to 8-bit gray-scale. *Border band removal* detects a rectangular bounding box surrounded by a maximal homogeneous image region adjacent to image borders (border band). If the resulting bounding box is empty or smaller than `elim_area` or if maximal recursion depth has been reached, an empty bounding box is returned, terminating recursion. The *separator line detection* modules are invoked separately for vertical and horizontal directions, so they deal with a single direction θ and return a list of corresponding separator lines. An empty list is returned if the respective image dimension (width or height) is smaller than `mindim` or if no separator lines are found. If the returned lists for both directions are empty, recursion is terminated and the current image (without border bands) is returned. The *decision about vertical or horizontal separation* is trivial if one of both lists of separator lines is empty. Otherwise the decision is made based on the regularity of separator distances:

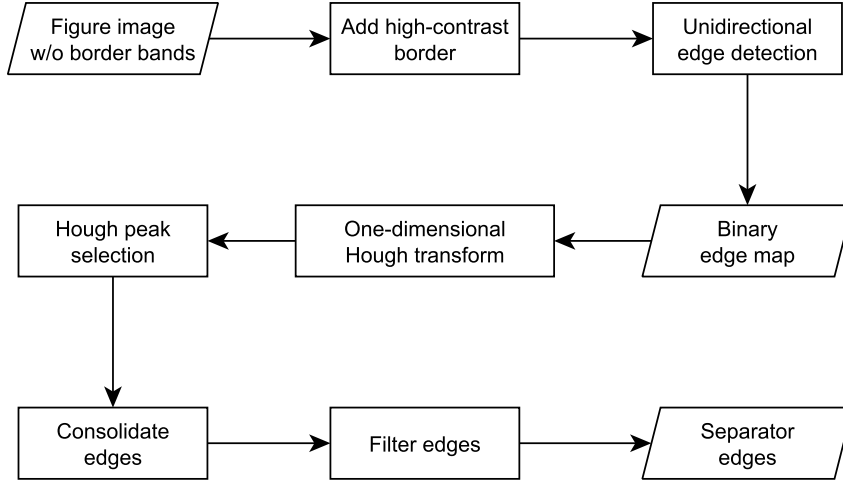


Fig. 3 Edge-based separator line detection.

locations of separator lines and borders are normalized to the range $[0,1]$, and the direction (vertical or horizontal) yielding the lower variance of adjacent distances is chosen. Finally, the current figure image is divided into subimages along the chosen separation lines, and the algorithm is applied recursively to each subimage.

2.2.3 Edge-based Separator Detection

The edge-based separator line detection algorithm aims at detecting full-length edges of a certain direction θ (vertical or horizontal) in a given gray-scale image. It comprises the following processing steps depicted in Fig. 3: (1) unidirectional edge detection, (2) peak selection in one-dimensional Hough transform, and (3) consolidation and filtering of candidate edges.

Edge detection is implemented by a one-dimensional Sobel filter and subsequent thresholding (`edge_sobelthresh`) to produce a binary edge map. The one-dimensional Hough transform counts the number of edge points aligned on each line in direction θ . So the peaks correspond to the longest edges, and their locations identify candidate separator edges. To make borders appear as strong Hough peaks, we add an artificial high-contrast border to the image prior to edge detection. Peaks are identified by an adaptive threshold t that depends on the recursion depth k (zero-based), the maximal value m of the current Hough transform, and the fill ratio f of the binary edge map (fraction of non-zero pixels, $0 \leq f \leq 1$), see (Eq. 1). α and β are internal parameters (`edge_houghratio_min` and `edge_houghratio_base`).

$$h = \alpha * \beta^k, \quad t = m * \left(h + (1 - h) * \sqrt{f} \right). \quad (1)$$

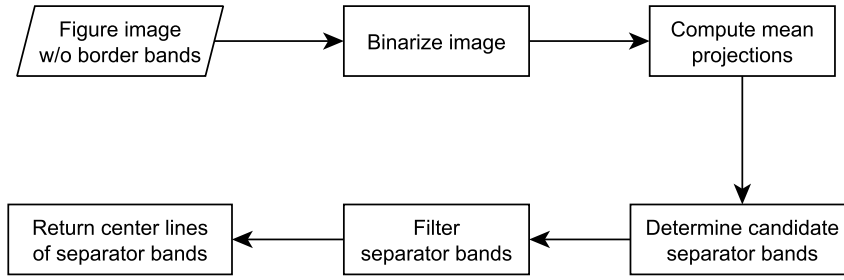


Fig. 4 Band-based separator line detection.

The rationale behind these formulas is to cope with noise in the Hough transform. Hough peaks were observed to become less pronounced as image size decreases (implied by increasing recursion depth) and as the fill ratio f increases (more edge points increase the probability that they are aligned by chance). Equation (1) ensures a higher threshold in these cases. Additionally, as recursion depth increases, the algorithm should detect only more pronounced separator edges, because further figure subdivisions become less likely.

Hough peak selection also includes a similar regularity criterion as used for deciding about vertical or horizontal separation (see Section 2.2.2): the list of candidate peaks is sorted by their Hough values in descending order, and candidates are removed from the end of the list until the variance of normalized edge distances of remaining candidates falls below a threshold (`edge_maxdistvar`). Candidate edges resulting from Hough peak selection are then consolidated by filling small gaps (of maximal length given by `edge_gapratio`) between edge line segments (of minimal length given by `edge_lenratio`). Finally, edges that are too short in comparison to image height or width (threshold `edge_minseplength`), or too close to borders (threshold `edge_minborderdist`) are discarded.

2.2.4 Band-based Separator Detection

The band-based separator detection algorithm aims at locating homogeneous rectangular areas covering the full width or height of the image, which we call *separator bands*. Since this algorithm is intended primarily for gray-scale illustration images with light background, we assume that separator bands are white or light gray. The algorithm consists of four steps illustrated in Fig. 4: (1) image binarization, (2) computation of mean projections, (3) identification and (4) filtering of candidate separator bands.

Initially, we binarize the image using the mean intensity value as a threshold. We then compute mean projections along direction θ (vertical or horizontal), that is, the mean value of each line of pixels in this direction. A resulting mean value will be 1 (white) if and only if the corresponding line contains only white pixels. Candidate separator bands are then determined by

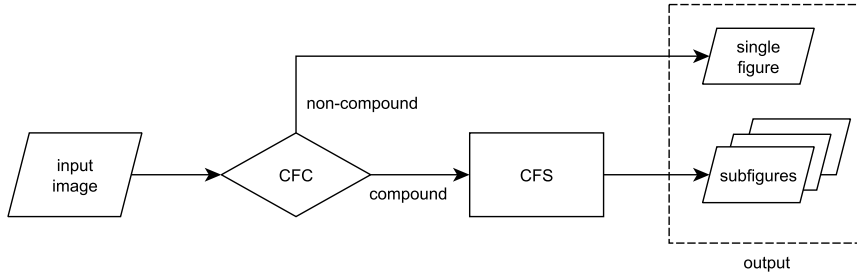


Fig. 5 Process chain consisting of compound figure classifier (CFC) and compound figure separation (CFS).

identifying maximal runs of ones in the vector of mean values that respect a minimal width threshold (`band_minseppwidth`). They are subsequently filtered using a regularity criterion similar to Hough peak selection (see Section 2.2.3), this time using distance variance threshold `band_maxdistvar`. Finally, selected bands that are close to the image border (threshold `band_minborderdist`) are discarded, and the center lines of remaining bands are returned as separator lines.

2.3 Chained Classification and Separation

Processing compound figures in a collection of scientific articles is expected to happen in a two-stage process as illustrated in Fig. 5: (1) all article images are classified as *compound* or *non-compound* by applying a compound figure classifier (CFC); (2) the predicted *compound* images are then processed by a compound figure separation (CFS) algorithm to obtain subfigures. The resulting set of subfigures and predicted *non-compound* figures can then be used for further application-specific processing (e.g. content-based indexing for retrieval). We are therefore interested in evaluating and improving the effectiveness of the *CFC-CFS process chain*, i.e. the quality of obtained subfigures and non-compound figures with respect to a gold standard and evaluation procedure (see Section 3.2).

Our proposed method for evaluating the effectiveness of the CFC-CFS chain will be described in Section 3.2. A guiding principle for improving the CFC-CFS chain is derived from consideration of the loss of effectiveness caused by different types of CFC errors: *false negatives* (compound figures classified as non-compound) may result in a larger loss than the same number of *false positives* (non-compound figures classified as compound), because false negatives are not processed by CFS and hence all contribute to the loss of effectiveness. On the other hand, there is a chance that false positives are not divided into subfigures by CFS (because it does not detect separation lines), and such instances of false positives will not degrade effectiveness of the CFC-CFS chain. Effectiveness can therefore be optimized on a validation set by biasing CFC

decisions towards the *compound* class. However, this is easy to achieve only for CFC algorithms that deliver predicted class probabilities, like logistic regression, but not for SVM.

The different importance of misclassifications of a binary classifier depending on true classes can be expressed by a 2×2 misclassification loss matrix (Eq. (2)) [4]. Rows correspond to true classes and columns to predicted classes, where in the case of CFC the first row or column is assigned to class *non-compound* (C_0) and the second row or column to class *compound* (C_1). The entries of loss matrix (Eq. (2)) denote the fact that misclassification of true compound figures incurs a loss that is by a factor of α larger than that of misclassification of true non-compound figures (if $\alpha > 1$). If the classifier is able to predict class probabilities $p(C_k|x)$ for a given image x , the decision of the classifier can be optimized with respect to expected misclassification loss $E_k(x)$ (Eq. (3)): image x is assigned to class C_k that minimizes $E_k(x)$ ($k = 0$ or $k = 1$). For the special form of loss matrix given in (Eq. (2)), this criterion reduces to a simple threshold on class probability $p(C_1|x)$: image x is assigned to class C_1 if and only if Eq. (4) holds. The parameter α can be selected by optimizing effectiveness of the CFC-CFS process chain on a validation set.

$$L = \begin{pmatrix} 0 & 1 \\ \alpha & 0 \end{pmatrix} \quad (2)$$

$$E_k(x) = \sum_i L_{ik} p(C_i|x) \quad (3)$$

$$p(C_1|x) \geq \frac{1}{1 + \alpha} \quad (4)$$

3 Experiments and Results

We evaluate our approach on separate datasets for CFC, CFS, and the CFC-CFS process chain, which are described in Section 3.1. As there is no agreement on a standard evaluation protocol for CFS in the research community yet, we use two different evaluation procedures, described in Section 3.2. Additionally, we propose to slightly extend existing CFS evaluation protocols in order to apply them to CFC-CFS chains. Evaluation results for CFC, CFS, and CFC-CFS chain are presented and discussed in Sections 3.3, 3.4, and 3.5, respectively.

3.1 Datasets

We used several datasets to train and evaluate the different components of our approach in our experiments. All of them were derived from the dataset of about 75,000 biomedical articles used for ImageCLEF medical tasks since 2012 [12]. Those articles were retrieved from PubMed Central² by selecting open

² <http://www.ncbi.nlm.nih.gov/pmc/>

Table 2 Datasets used in our experiments. CFC = compound figure classification, CFS = compound figure separation, MC = modality classification; CO = compound, ILL = illustration.

Dataset	Training		Test	
	Images	Annotations	Images	Annotations
ImageCLEF CFC	10387	6121 CO (59%)	10434	6144 CO (59%)
ImageCLEF CFS	3403	14531 subfigures	3381	12789 subfigures
NLM CFS			380	1656 subfigures
ImageCLEF MC first	1071	607 ILL (57%)	497	261 ILL (53%)
ImageCLEF MC majority	895	514 ILL (57%)	428	243 ILL (57%)
ImageCLEF MC unanimous	867	508 ILL (59%)	398	226 ILL (57%)
ImageCLEF MC greedy	1071	712 ILL (66%)	497	325 ILL (65%)
CFC-CFS	6806	17934 subfigures	6752	16154 subfigures

access journals that allow for free redistribution of data. The articles of the ImageCLEF dataset contain about 300,000 images of unconstrained modalities (biomedical images, diagrams, charts, photographs, etc.) and subfigure structure (*compound* and *non-compound* images).

A subset of about 21,000 images used for the ImageCLEF 2015 medical tasks [11] formed the basis for most datasets used in our experiments, namely all datasets labeled *ImageCLEF* in Table 2. The CFC training dataset provided by ImageCLEF task organizers contained some erroneous samples (23 images had contradicting annotations), which have been removed from the training set. Table 2 refers to the cleaned CFC training set only. The CFC dataset consists of 59% compound images (CO), both in training and test subsets, providing reasonable conditions for training and evaluating a binary classifier. A similar split of classes is present in the modality classification (MC) datasets, which are used to train and evaluate the binary classifier for illustrations (ILL) (see Section 2.2.1).

The MC datasets were derived from the dataset of the ImageCLEF 2015 multi-label image classification task [11]. The images are provided with one or more labels of 29 classes (organized in a class hierarchy), which have been mapped to two meta classes: the *illustration* meta class comprises all “general biomedical illustration” classes except for chromatography images, screen-shots, and non-clinical photos. These classes and all classes of diagnostic images have been assigned to the *non-illustration* meta class. About 36% of the images in the training set are labeled with multiple classes, corresponding to compound images. Training and evaluation of the illustration classifier (Section 2.2.1) requires mapping the set of labels of a given image to a single meta class. We implemented four mapping strategies that first assign each image label to the *illustration* or *non-illustration* meta class, and then operate differently on the list L of meta labels associated with a given image: (1) the *first* strategy simply assigns the first meta label of L to the image; (2) the *majority* strategy selects the meta label occurring most often in L , dropping the image from the dataset if both meta labels occur equally often; (3) the *unanimous* strategy only assigns a meta label to the image if all meta labels in L are equal, otherwise the image is dropped from the dataset; and (4) the

greedy strategy maps an image to the *illustration* label if L contains at least one such meta label, otherwise the image is assigned the *non-illustration* label. Note that *majority* and *unanimous* strategies discarded up to 20% of images in the original dataset. Whereas *majority* and *unanimous* mapping strategies are expected to improve classification accuracy, the *greedy* strategy aims at increasing CFS effectiveness based on the assumption that a compound image containing an illustration subfigure is more likely to have separator bands than separator edges.

A research group at the U.S. National Library of Medicine (NLM) had created a dataset to evaluate their CFS approach (and related algorithms) [1] well before the first CFS task at ImageCLEF was issued in 2013. This dataset contains 400 images and 1764 ground-truth subfigures and hence is substantially smaller than the ImageCLEF CFS test dataset. Moreover, it shares 20 images with the training set and 27 images with the test set of the ImageCLEF CFS dataset. The reason for the non-empty intersection of these datasets is that the NLM dataset was sampled from a set of 231,000 article images used at ImageCLEF 2011, which was extended later to the ImageCLEF dataset mentioned at the beginning of this section. Since we used the ImageCLEF CFS training set for parameter optimization, we removed the 20 images in the intersection from the NLM dataset for our experiments. The resulting reduced dataset is listed in Table 2 as *NLM CFS* dataset.

For evaluation of the CFC-CFS process chain, we extended the ImageCLEF CFS test dataset (3381 images) with the same number of non-compound images sampled at random from the ImageCLEF CFC test dataset. After removing five images that occurred in both portions of this dataset³, a test dataset with 6752 images was obtained. In a similar manner, a validation set of 6806 images was constructed from ImageCLEF CFS and CFC training datasets (appearing as “training set” in the last line of Table 2). Non-compound images of the CFC-CFS dataset were annotated with a single subfigure covering the entire image, as explained in Section 3.2.2.

3.2 Evaluation Methods

While evaluation of classification algorithms is a well-studied problem [5, 9, 14, 16, 24], evaluation of compound figure separation has been addressed by two different ad-hoc procedures only [1, 10]. Both evaluation procedures first determine which detected subfigures of a given compound image are correct (*true positive*) with respect to ground-truth subfigures, and then compute an evaluation measure from the number of true positive subfigures over the dataset. However, the way by which true positive subfigures are determined, and which

³ Ideally, the intersection should be empty, because the CFS dataset should contain only compound images. However, manual inspection of images in the intersection revealed that both CFS and CFC datasets contain errors and that the distinction between compound and non-compound images is not always clear.

evaluation measures are calculated, differs between the two proposed evaluation procedures, which are described in detail in the following section. In Section 3.2.2 we propose to also apply CFS evaluation methods to measure the effectiveness of the CFC-CFS process chain.

3.2.1 CFS Evaluation

To describe the evaluation protocols in detail, we introduce the following notation. Without loss of generality, we assume that a subfigure is represented by rectangular area R (bounding box) within an image, and denote its area size (number of contained pixels) by $|R|$. For a given compound figure, let $\{G_i | i \in I\}$ be the set of ground-truth subfigures, and $\{F_j | j \in J\}$ the set of subfigures detected by the CFS algorithm that should be evaluated. Note that the overlap area $G_i \cap F_j$ between subfigures is again a rectangle (or empty). The two evaluation protocols employ different definitions of the *overlap ratio* between G_i and F_j , given in Equations (5) and (6). ρ_{ij}^G is the overlap ratio with respect to ground-truth subfigure G_i , ρ_{ij}^F calculates the ratio with respect to detected subfigure F_j .

$$\rho_{ij}^G = \frac{|G_i \cap F_j|}{|G_i|} \quad (5)$$

$$\rho_{ij}^F = \frac{|G_i \cap F_j|}{|F_j|} \quad (6)$$

The evaluation procedure used for ImageCLEF CFS tasks [10] iterates over ground-truth subfigures G_i and, for a given G_i , looks for a detected subfigure F_j with maximal overlap ρ_{ij}^F . F_j is associated with G_i if $\rho_{ij}^F > 2/3$ and if F_j has not already been associated with a different ground-truth subfigure. The result is a set of one-to-one associations between ground-truth subfigures and detected subfigures, which are regarded as true positives. Note that although the set of associations may depend on the order of iterations over G_i , the number C of these associations does not. Accuracy can therefore be defined per compound figure as $C / \max(N_G, N_D)$, where N_G and N_D are the numbers of ground-truth and detected subfigures, respectively. Accuracy on the test set is the average of accuracy values computed for each compound figure.

The authors of the NLM CFS dataset [1] (see Section 3.1) used a different criterion to determine true positive subfigures. A detected subfigure F_j is considered true positive if and only if there is a ground-truth subfigure G_i with $\rho_{ij}^G > 0.75$ and $\rho_{kj}^G < 0.05$ for all other ground-truth subfigures G_k . That is, subfigure F_j has a notable overlap with one ground-truth subfigure only. Given the total number N of ground-truth subfigures in the dataset, the total number D of detected subfigures, and the number T of detected true positive subfigures, the usual definitions for classifier evaluation measures can be applied to obtain precision P , recall R , and F_1 measure, see Eq. (7). Note that accuracy is not well-defined in this setting, because the number of negative results (not detected arbitrary bounding boxes) is theoretically unlimited.

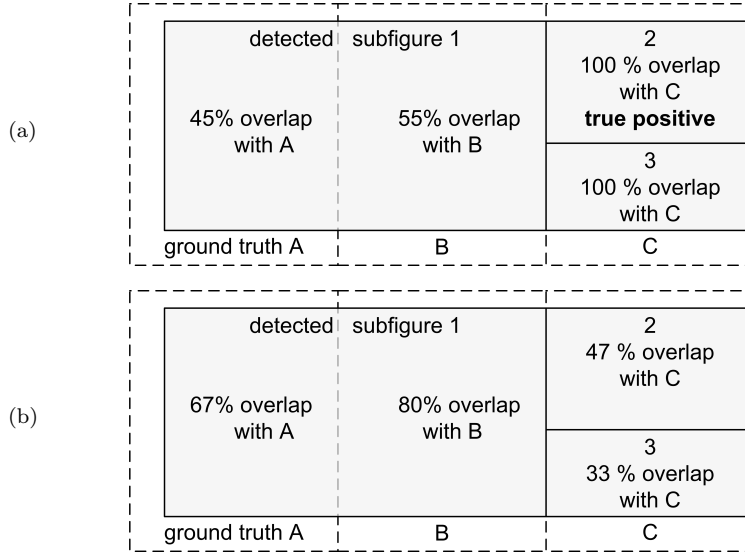


Fig. 6 Determination of true positive detected subfigures by (a) ImageCLEF and (b) NLM CFS evaluation procedures.

$$P = \frac{T}{D} , \quad R = \frac{T}{N} , \quad F_1 = \frac{2 * P * R}{P + R} . \quad (7)$$

Figure 6 illustrates two different ways of determining *true positive* detected subfigures for an example compound figure, which consists of three ground-truth subfigures: A, B, and C. We assume that a hypothetical CFS algorithm, given this compound figure as input, produced three subfigures – indicated as subfigures 1, 2, and 3 – at its output. In Figure 6, the resulting detected subfigures appear on the foreground, partially overlapping the three ground-truth subfigures in the background. The ImageCLEF and NLM evaluation protocols for this case will result in two different assessments, as follows:

- Figure 6 (a): The ImageCLEF evaluation procedure considers only one of subfigures 2 or 3 as true positive, depending on which of them gets associated first with C. Note that both overlap ratios ρ_{C2}^F and ρ_{C3}^F are 100%. Subfigure 1, however, is regarded as false positive, because its overlap ratio, according to definition (6), with any ground-truth subfigure does not exceed $2/3$. The resulting accuracy is therefore $1/3$, since only one of the three detected subfigures qualifies as true positive.
- Figure 6 (b): The NLM evaluation procedure, on the other hand, determines that all detected subfigures should be considered false positives, because for subfigures 2 and 3 the overlap ratio, according to definition (5), with any ground-truth subfigure is too small (i.e., less than 75%), and subfigure 1 overlaps with two ground-truth subfigures (A and B) by at least 5%.

3.2.2 CFC-CFS Chain Evaluation

We propose to apply the CFS evaluation methods described in the previous section to the output of the CFC-CFS process chain (Section 2.3). Because CFS test datasets contain only compound figures, but the dataset for CFC-CFS chain evaluation also includes non-compound figures (Section 3.1), we need to extend CFS evaluation procedures by a convention to represent non-compound figures. We adopt the obvious solution to consider non-compound figures as “compound figures with a single subfigure” and represent each of them by a bounding box covering the entire image. This extension needs to be implemented in three different places of the evaluation procedure: (1) for ground-truth annotation, (2) for images classified as *non-compound* by CFC, and (3) for images classified as *compound* that are not divided into subfigures by CFS (because it does not detect proper separator lines).

Unmodified CFS evaluation algorithms can then be applied to the output of the CFC-CFS chain. Note that the ImageCLEF evaluation algorithm will assign 100% accuracy for true non-compound images only if there is exactly one “detected” subfigure in the CFC-CFS output, no matter what the bounding boxes are. Similarly, the NLM evaluation algorithm will find at most one true positive subfigure in a true non-compound image, but in this case the area of the “detected” bounding box is relevant (it must cover at least 75% of the entire image).

3.3 Compound Figure Classifier

We used the ImageCLEF CFC dataset (Section 3.1) to train and evaluate the various combinations of feature sets and classifier algorithms described in Section 2.1. More specifically, we trained all three classifiers on 40 feature sets created by instantiating the 10 feature sets listed in Table 1 for four values of k (4, 8, 16, and 32). The quantization parameters were kept constant as $p = 5$, $q = 8$, and $h = 3$, as these values gave good classification performance in preliminary experiments. To enable a fair comparison with SVM, the logistic regression classifier used a probability threshold of 0.5, corresponding to a symmetric misclassification loss matrix (Eq. (2)) with $\alpha = 1$.

Results of CFC experiments are presented in Table 3. From the 120 combinations of classifier algorithm, feature sets, and number k of spatial bins that were tested in experiments, we report only the best three and the worst results – separated by a dashed line in Table 3 – for each classifier algorithm with respect to accuracy.

Results indicate that feature set 434 achieves good classification performance for all three tested classifier algorithms with a rather low dimensionality of 96 (see Table 1). Feature set 134 (with 224 dimensions) with $k = 16$ spatial bins showed the same accuracy (76.9%) for both linear classifiers, becoming the best overall performer in both cases. The surprisingly low classification performance of kernel SVM is probably due to underfitting caused by default

Table 3 Evaluation results of compound figure classifier on ImageCLEF CFC test set. From the 120 tested combinations of classifier algorithm, feature set, and number k of spatial bins, only the best three and the worst result for each classifier algorithm are reported. LogReg = logistic regression, SVM = support vector machine.

Classifier	Feature Set	k	Accuracy%	FP%	FN%
LogReg	134	16	76.9	16.9	6.2
LogReg	434	8	76.6	18.2	5.2
LogReg	434	16	76.6	17.7	5.7
LogReg	011	4	61.3	8.5	30.2
linear SVM	134	16	76.9	14.6	8.6
linear SVM	434	8	76.8	16.6	6.7
linear SVM	434	16	76.5	15.9	7.6
linear SVM	222	4	63.9	25.9	10.2
kernel SVM	034	4	75.5	20.4	4.1
kernel SVM	444	4	75.3	20.8	3.9
kernel SVM	434	4	74.2	23.0	2.9
kernel SVM	666	32	59.0	41.0	0.0

SVM hyperparameters; both box constraint C and standard deviation σ of the radial basis function (RBF) kernel were kept at the default value 1.

Remarkably, the false positive rate of all well-performing classifiers in Table 3 is systematically higher than the false negative rate. This can be explained by two possible causes: first, the training set is slightly imbalanced (59% compound images), which may cause the classifier to decide in favor of the *compound* class in uncertain cases; second, the feature sets used for CFC produce a denser spatial distribution of non-compound images in the feature space than for compound ones, reinforcing the imbalanced training effect. In fact, the CFC features described in Section 2.1 have been designed to capture the existence of separators between subfigures. If such separators do not exist, feature values may exhibit a low variance across different images.

Compared to the best CFC run using visual-only features submitted to ImageCLEF 2015 by Wang et al. [28], which achieved 82.8% accuracy on the same dataset, our results are inferior by a margin of about 6%. However, as the approach of Wang et al. essentially employs a CFS algorithm (connected component analysis and band separator detection), we suppose that our CFC method has significant advantages with respect to efficiency for online classification. Extraction of the 111 feature set, which is the most complex of our proposed feature sets, took 81 milliseconds per image on average (excluding reading the image file from disk) using a MATLAB implementation on an Intel E8400 CPU operated at 3 GHz. This execution time corresponds to a processing rate of 12.3 images per second.

3.4 Compound Figure Separation

Our CFS approach is evaluated mainly on the ImageCLEF CFS dataset (Section 3.1) using the ImageCLEF evaluation procedure (Section 3.2). The internal parameters of our CFS algorithm, including implementation options of

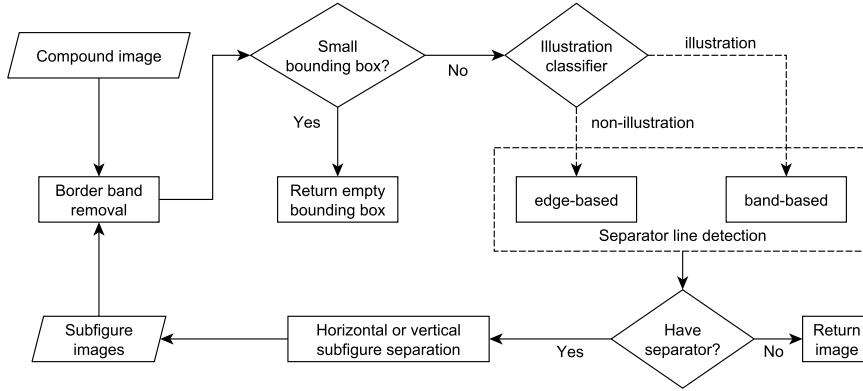


Fig. 7 Variant of proposed CFS algorithm that applies the illustration classifier to every detected subfigure prior to splitting it further.

the illustration classifier, are optimized on the training portion of the dataset as described in Section 3.4.1, prior to evaluating CFS performance on the test dataset. To analyze the effectiveness of the illustration classifier for CFS, we also report results for different classifier implementation options obtained by keeping these options constant during parameter optimization.

Moreover, we consider a variant of the proposed CFS algorithm in which the illustration classifier has been replaced by a binary random decision unit, which predicts that a given input image is an *illustration* with probability p . For $p = 0$, the CFS algorithm will always use edge-based separator detection, and for $p = 1$ band-based separator detection will be applied to every input image. The rationale for choosing p as the actual *illustration* decision rate of the classifier on the test dataset is to allow a fair comparison between the “random decision” variant and the proposed CFS algorithm, which should allow us to quantify the utility of the illustration classifier in our CFS approach.

The proposed CFS algorithm applies the illustration classifier once to each input image and reuses the classifier’s decision in all recursive invocations of the separator detection module (see Fig. 2). To answer the question whether applying the classifier anew for each recursive invocation improves CFS performance, we also consider this algorithmic variant in our experiments, depicted in Fig. 7.

To enable comparison with other CFS approaches in the literature, we further evaluate our approach on the NLM dataset using the evaluation procedure proposed by its authors (see Section 3.2). By using the same parameter values obtained by optimization on the ImageCLEF training set, CFS results on the NLM dataset provide additional information about the generalization ability of our CFS algorithm.

3.4.1 Parameter Optimization

The proposed CFS algorithm takes 17 internal parameters listed in Table 4. Initial parameter values were chosen manually by looking at the results produced for a few training images. They were used during participation in ImageCLEF 2015 [26]. For parameter optimization, the CFS algorithm was evaluated for various parameter combinations on the ImageCLEF 2015 CFS training dataset (3,403 compound images, 14,531 ground-truth subfigures) using the evaluation tool provided by ImageCLEF organizers. Due to the number of parameters and the run time of a single evaluation run (about 17 minutes), a grid-like optimization evaluating all possible parameter combinations in a certain range was not feasible. Instead, we applied a hill-climbing optimization strategy to locate the region of a local maximum and then used grid optimization in the neighborhood of this maximum.

More precisely, we defined up to five different values per parameter, including the initial values, on a linear or logarithmic scale, depending on the parameter. Then a set of parameter combinations was generated where only one parameter was varied at a time and all other parameters were kept at their initial values, resulting in a feasible number of parameter combinations to evaluate (linear in the number of parameters). After measuring accuracy on the training set, the most effective value of each parameter was chosen as its new *optimal* value. For parameters whose optimal values differed from the initial ones, the range was centered around the optimal value. Other parameters were fixed at their latest value. The procedure was repeated until accuracy improved by no more than 5%, which happened after three iterations. Finally, after sorting parameter combinations by achieved accuracy, the five most effective parameters were chosen for grid optimization, where only two “nearly optimal” values (including the latest optimal value) per parameter were selected.

The effect of parameter optimization was surprisingly strong: whereas the initial parameter configuration achieved an accuracy of 43.5% on the training set, performance increased to 84.5% after hill-climbing optimization, and finished at 85.5% after grid optimization.

3.4.2 Evaluation on ImageCLEF Dataset

Experimental results are shown in Table 5. For comparison, we also included a previous version of our approach [26] that did not use optimized parameters, and the best approach submitted to ImageCLEF 2015 (by NLM). We evaluated the proposed algorithm with optimized parameters (see Section 3.4.1) and with different implementations and feature sets for the illustration classifier, as described in Section 2.2.1. Because logistic regression using *simple2* features was found to be most effective by parameter optimization when trained on the *greedy* set, we focused on this training set when evaluating other classifier implementations. Internal SVM parameters were optimized on the entire ImageCLEF 2015 multi-label classification test dataset (see Section 3.4.4) to

Table 4 Internal parameters of proposed CFS algorithm. Initial parameter values were used during ImageCLEF 2015 participation [26], optimal values were obtained by parameter optimization on the ImageCLEF 2015 CFS training dataset. Parameters marked by * use units of image width, height, or area, depending on the parameter and processing direction (horizontal or vertical).

Parameter	Initial	Optimal	Meaning
Main algorithm			
classifier_model	first	greedy	first, majority, unanimous, or greedy (see Section 2.2.1)
decision_threshold	0.5	0.1	minimal illustration class probability to decide in favor of band-based separator detection
mindim	50	200	minimal image dimension (pixels) to apply separator detection to
elim_area	0	0.03	area threshold to eliminate small bounding boxes*
Edge-based separator detection			
edge_maxdepth	10	10	maximal recursion depth
edge_sobelthresh	0.05	0.02	threshold for Sobel edge detector
edge_houghratio_min	0.25	0.2	minimal ratio of Hough values for peak selection
edge_houghratio_base	1.2	1.5	base of recursion depth dependency for Hough peak selection
edge_maxdistvar	0.0001	0.1	maximal variance of separator distances for regularity criterion*
edge_gapratio	0.2	0.3	gap threshold for edge filling*
edge_lenratio	0.05	0.03	length threshold for edge filling*
edge_minseplength	0.7	0.5	minimal separator length*
edge_minborderdist	0.1	0.05	minimal distance of separators from border*
Band-based separator detection			
band_maxdepth	2	4	maximal recursion depth
band_minseppwidth	0.03	0.0001	minimal width of separator bands*
band_maxdistvar	0.0003	0.2	maximal variance of separator distances for regularity criterion*
band_minborderdist	0.1	0.01	minimal distance of separators from border*

maximize classification accuracy. The optimized `decision_threshold` parameter for deciding between edge-based and band-based separator detection is effective only for logistic regression classifiers, because SVM predictions do not provide class probabilities. To confirm the effectiveness of the illustration classifier, we also included results for algorithm variants where the classifier has been replaced by a random decision selecting band-based separator detection with probability p . The value $p = 0.741$ corresponds to the decision rate of the most effective classifier (LogReg,simple11,greedy). $p = 0$ and $p = 1$ represent algorithms that always use edge-based or band-based separator detection, respectively. Finally, the algorithm variant *SubfigureClassifier* applies the illustration classifier not only once per compound image, but also to each subimage during recursive figure separation (see Fig. 7).

Table 5 Experimental results on the ImageCLEF 2015 CFS test set. Illustration classifiers are described in Section 2.2.1 (LogReg = logistic regression). BB denotes the percentage of images (or decisions*) where band-based separator detection was applied.

Algorithm	Classifier	BB %	CFS Accuracy %
Previous [26]	LogReg,simple2,first		49.4
NLM [20]	manual	95.7	84.6
Proposed	LogReg,simple2,first	61.6	84.2
Proposed	LogReg,simple2,majority	61.1	84.1
Proposed	LogReg,simple2,unanimous	61.8	84.2
Proposed	LogReg,simple2,greedy	75.8	84.8
Proposed	LogReg,simple11,greedy	74.1	84.9
Proposed	SVM,simple2,greedy	58.6	83.5
Proposed	SVM,simple11,greedy	60.3	83.5
Proposed	SVM,CEDD,greedy	59.2	82.8
Proposed	SVM,CEDD,simple11,greedy	59.6	83.2
Proposed	random,p=0.741	74.7	75.4
Proposed	no classifier,p=0	0	58.0
Proposed	no classifier,p=1	100	82.2
SubfigureClassifier	LogReg,simple11,greedy	60.1*	84.0

When comparing our results to NLM’s approach, we note that the authors of [20] manually classified the test set into stitched (4.3%) and non-stitched (95.7%) images, whereas our approach uses automatic classification. Using band-based separator detection for all test images (no classifier, $p = 1$) works surprisingly well (82.2% accuracy), which can be explained by the low number of stitched compound images in the test set. On the other hand, using edge-based separator detection for all test images (no classifier, $p = 0$) results in modest performance (58% accuracy), which we attribute to a significant number of subfigures without rectangular borders (illustrations) in the test set. Selecting edge-based or band-based separator detection using the illustration classifier improved accuracy for all tested classifier implementations. In fact, it turned out to be effective to bias the illustration classifier towards band-based separator detection and apply edge-based separator detection only to high-confidence non-illustration images. This happened in two ways: by using the *greedy* training set, and by optimizing the `decision.threshold` parameter for the logistic regression classifier. This explains why best results were obtained by logistic regression classifiers trained on the *greedy* training set.

To further analyze the effectiveness of separator detection selection, we partitioned the CFS test dataset into two classes according to decisions of the most effective CFS algorithm variant (LogReg,simple11,greedy) and evaluated detection results of this algorithm separately on the two partitions. Resulting accuracy values of 85.7% on the edge-based partition and 84.6% on the band-based partition show that the classifier was successful in jointly optimizing detection performance for both separator detection algorithms.

Our algorithm was implemented in MATLAB and executed on a PC with 8 GB RAM and an Intel E8400 CPU running at 3 GHz. The average total processing time per compound image was 0.3 seconds when an illustration

Table 6 Evaluation results on the NLM CFS dataset [1]. Precision (P), recall (R), and F_1 score are computed from the total number of ground-truth (G), detected (D), and true positive (T) subfigures.

Algorithm	G	D	T	P%	R%	F_1 %
Proposed (LogReg)	1656	1550	1314	84.8	79.4	82.0
Proposed (SVM)	1656	1584	1297	81.9	78.3	80.1
Apostolova et al. [1]	1764	1482	1276	86.1	72.3	78.6

classifier with *simple* features was used, and 0.9 seconds when a classifier with CEDD features was applied. Note that the efficiency of other known approaches in the literature is either not documented [1] or by an order of magnitude lower ([8] reported 2.4 seconds per image).

3.4.3 Evaluation on NLM Dataset

Table 6 shows the results of evaluating our proposed algorithm on the NLM CFS dataset (see Section 3.1) using the NLM evaluation procedure described in Section 3.2. We used the same parameter settings as in Section 3.4.2 to demonstrate the generalization capability of our algorithm. We selected the most effective illustration classifiers using logistic regression and SVM, respectively. They both use *simple11* features and the *greedy* training set. For convenience, we also included the results reported in [1] for a direct comparison with our approach.⁴

Results show that the relative performance of the proposed algorithm using different classifiers is consistent with evaluation results in Section 3.4.2. The proposed algorithm could detect 10% more true positive subfigures than the image panel segmentation algorithm of Apostolova et al. [1], leading to a higher recall rate. On the other hand, precision is only slightly lower. Note that algorithm [1] has been used as a component in NLM’s CFS approach [20] referenced in Section 3.4.2.

3.4.4 Illustration Classifier Accuracy

To investigate the correlation of illustration classifier performance and effectiveness for CFS, we evaluated classification accuracy for the various classifier implementations considered in Section 3.4.2 on the test dataset of the Image-CLEF 2015 multi-label image classification task [11]. Labels of test images were mapped to binary meta classes using the same procedure as described in Section 2.2.1, resulting in 497 images for *first* and *greedy* test sets, 428 images for *majority*, and 398 images for *unanimous* test set. Evaluation results are shown in Table 7. The decision threshold for logistic regression was set to

⁴ The dataset reported in [1] contains 400 images with 1764 ground-truth subfigures, so reported recall may be up to 0.4% higher if evaluated on the 398 images of the dataset available to us.

Table 7 Classification accuracy on ImageCLEF 2015 multi-label image classification test dataset (497 images) for different implementation options of illustration classifier. Features and training sets are described in Section 2.2.1, LogReg = logistic regression.

Classifier	Features	Training Set	Accuracy %
LogReg	simple2	first	82.5
LogReg	simple2	majority	86.5
LogReg	simple2	unanimous	88.2
LogReg	simple2	greedy	84.7
LogReg	simple2	greedy	84.7
LogReg	simple11	greedy	83.7
SVM	simple2	greedy	84.3
SVM	simple11	greedy	84.3
SVM	CEDD	greedy	87.1
SVM	CEDD_simple11	greedy	86.7

0.5 to provide a fair comparison with SVM. Internal parameters of SVM (box constraint C and standard deviation σ of RBF kernel) were optimized using two-fold cross-validation on the test set.

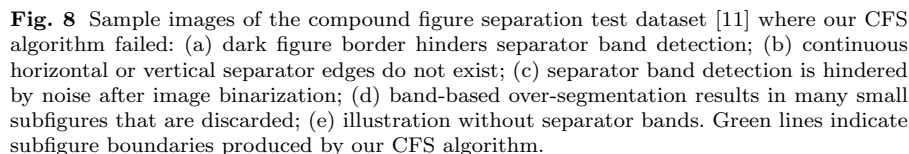
The upper part of Table 7 tells us that *majority* and *unanimous* training sets improve classification performance, although we know from Section 3.4.2 that this does not help CFS effectiveness. From the lower part of Table 7 we note that, interestingly, SVM does not perform better on *simple2* features than logistic regression and causes only a modest improvement (around 3%) on CEDD features (144-dimensional). This may indicate the need to select more discriminative features for this classification task in future work, although results of Section 3.4.2 suggest that accuracy of the illustration classifier is not a critical factor of the proposed CFS algorithm.

3.4.5 Limitations of CFS Algorithm

Figure 8 shows some examples of test images where our algorithm did not perform as expected, for different reasons. In parts (a) and (e), the separator band detection algorithm fails due to the presence of dark lines, around the image and/or within the compound figure (which, ironically, was probably drawn to serve as visual separator between the subfigures). In part (b), the edge-based separation algorithm fails due to the lack of *continuous* horizontal or vertical separator edges. In part (c), the presence of noise in the original image leads to an imperfect binarized version of the figure, which consequently impacts the performance of the separator band detection algorithm. Lastly, part (d) shows an example of over-segmentation, in which the proposed band-based algorithm first produces many subfigures, which are subsequently discarded for being too small.

3.4.6 Limitations of CFS Evaluation

The validity of a CFS evaluation procedure depends on both the quality of the test dataset, including ground-truth annotations, and the meaningfulness



of the adopted performance metric. We recognized limitations in both aspects during experiments on the ImageCLEF CFS dataset. From the 260 images of the test set that received an accuracy of zero after being processed by our best CFS run (see Table 5), we randomly selected 10 images and investigated the reason for failure. For three of them our CFS algorithm produced meaningful results, but errors in ground-truth annotations caused the ImageCLEF evaluation method to return accuracy 0, as illustrated for one of those images in Fig. 9.

Fig. 9.

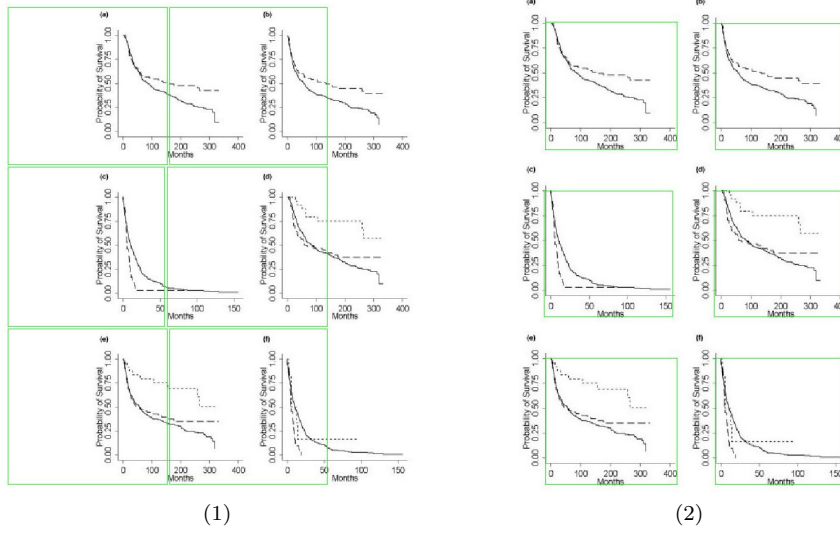


Fig. 9 (1) Example image of ImageCLEF CFS test dataset with imprecise ground-truth annotations. (2) Result produced by our CFS algorithm, which was erroneously determined as having accuracy 0.

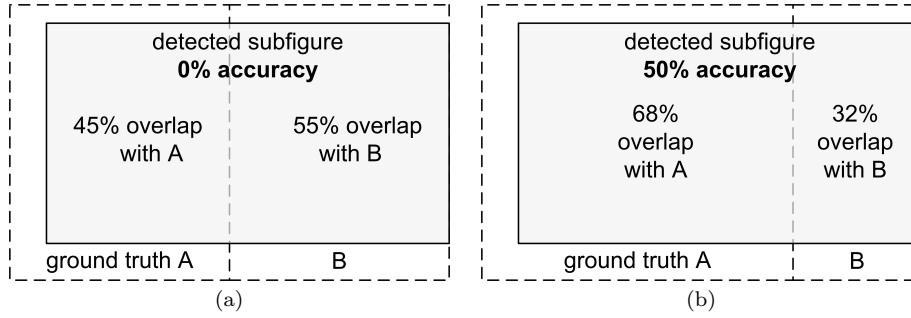


Fig. 10 Two similar under-segmentation cases lead to very different accuracy values according to ImageCLEF CFS evaluation procedure, because one of the ground-truth subfigures covers (a) less or (b) more than $2/3$ of the single detected subfigure.

In the second aspect, the ImageCLEF CFS evaluation procedure exhibits a notable instability with respect to under-segmentation: the CFS result in Fig. 10(a) is assigned an accuracy of zero, because none of the overlap ratios with the two ground-truth subfigures exceeds $2/3$. On the other hand, for the similar situation in Fig. 10(b) the obtained accuracy is 50%, because one of the ground-truth subfigures covers more than $2/3$ of the single detected subfigure. Note that this problem does not occur with the NLM CFS evaluation procedure.

Table 8 Evaluation results of CFC-CFS chain for different algorithms and decision thresholds of the compound figure classifier (CFC). Decision thresholds are applicable to the logistic regression (LogReg) classifier only. The threshold marked by * was found to be optimal on the validation set. CR is the percentage of images classified as compound. In addition to accuracy on the total test set, accuracy values on the subsets of predicted compound (C) and non-compound (NC) images are shown.

CFC	Threshold	CR%	Accuracy%		
			C	NC	Total
LogReg	0.20	84	84.7	94.7	86.4
LogReg	*0.35	74	84.9	90.8	86.5
LogReg	0.50	66	85.2	86.6	85.6
LogReg	0.65	56	85.9	81.1	83.8
linear SVM	–	61	85.6	82.1	84.2
kernel SVM	–	74	84.4	95.6	87.3
<i>none</i>	0	100	85.1	–	85.1
<i>ideal</i>		50	84.9	100	92.5

3.5 CFC-CFS Chain

We used the CFC-CFS test dataset and evaluation procedure described in Sections 3.1 and 3.2.2, respectively, to evaluate the effectiveness of the proposed CFC-CFS process chain. Results obtained using the ImageCLEF CFS evaluation method are presented in Table 8. For each of the three CFC algorithms (logistic regression, linear SVM, and kernel SVM) evaluated in Section 3.3, we applied the best-performing parameter settings according to Table 3. From these classifier algorithms, only logistic regression delivers predicted class probabilities, which allows to tune the effectiveness of the CFC-CFS chain by optimizing the decision threshold (Equation (4) in Section 2.3). Optimization was performed by evaluating CFC-CFS effectiveness on the CFC-CFS validation set for decision thresholds d in the range $0.2 \leq d \leq 0.7$ using a step size of 0.05. The optimal value was found as $d = 0.35$, corresponding to weight $\alpha = 1.86$ of the misclassification loss matrix (Eq. (2)). In Table 8, we report results for four different decision thresholds on the test set. The optimal threshold selected during optimization on the validation set (indicated by *) also delivers best performance on the test set, confirming that improved performance for decision thresholds $d < 0.5$ is not caused by overfitting the validation set.

Column CR (“compound rate”) of Table 8 shows the percentage of input images classified as *compound* by the different CFC implementations. Separate accuracy values on the portions of the test set classified as compound and non-compound, respectively, indicate a natural trend: accuracy increases with decreasing size of the class-specific subset. For logistic regression, the increase of accuracy on the non-compound subset for shrinking decision thresholds overcompensates the moderate loss on the compound subset, improving total accuracy. As the decision threshold approaches zero, however, the number of predicted non-compound images and hence their effect on total accuracy becomes too small, leading to the observed local maximum of total accuracy for decision threshold $d = 0.35$.

High CFC-CFS accuracy on the subset of predicted non-compound images can also be explained by a low false negative rate of CFC: false negatives are true compound images classified as non-compound, which are not sent through CFS processing and hence hurt effectiveness of the CFC-CFS chain. This explains the good performance of kernel SVM in Table 8, although kernel SVM achieved inferior accuracy in CFC experiments (Table 3). From the three tested CFC algorithms, kernel SVM happened to have the lowest false negative rate at the cost of a high false positive rate, leading to a similar effect as decreasing the decision threshold for logistic regression.

From a wider perspective, however, effectiveness of CFC in the CFC-CFS process chain is rather limited when compared to processing all images of the test dataset with CFS only (indicated by classifier *none* in Table 8). In fact, our CFC implementations could improve CFC-CFS chain effectiveness by 2% only, whereas an *ideal* CFC algorithm that reproduces ground-truth class annotations would increase total accuracy by more than 7%.

Finally we note that all pairwise differences of total accuracy values in Table 8, which are mean values of accuracies determined for every input image, are statistically significant except for the difference between the first two lines in the table (logistic regression with decision thresholds 0.2 and 0.35, respectively). Significance has been tested at the 5% significance level using a paired t-test.

4 Conclusions

In this paper we have proposed, implemented, tested, and evaluated a method to automatically classify and separate compound figures often found in scientific articles. The proposed method consists of two main steps: (i) a supervised compound figure classifier (CFC) discriminates between compound and non-compound figures using task-specific image features; and (ii) an image processing algorithm is applied to predicted compound images to perform compound figure separation (CFS). Combined, they are referred to as the *CFC-CFS process chain*, to emphasize the dependencies and relationships between the two main blocks.

We have also introduced novel image features for compound figure classification and demonstrated that they can be used to achieve state-of-the-art CFC performance using well-known classifier algorithms.

Moreover, we have demonstrated that the proposed CFS algorithm outperforms state-of-the-art automatic and semi-automatic CFS approaches on two recently published biomedical datasets.

Lastly, we have established a method to evaluate the effectiveness of the CFC-CFS process chain and applied it to optimize the misclassification loss of CFC for maximal effectiveness in the process chain.

Future work might include algorithmic refinements to the CFS approach to address limitations (such as those illustrated in Figure 8), as well as implementation and testing of additional features and classification algorithms

for CFC. When larger training datasets become publicly available, the use of deep learning techniques (e.g., convolutional neural networks) should also be considered.

Acknowledgements We thank Sameer Antani (NLM) and the authors of [1] for providing their compound figure separation dataset for evaluation purposes. We are grateful to Laszlo Böszörményi (ITEC, AAU) for valuable discussions and comments on this work.

References

1. Apostolova, E., You, D., Xue, Z., Antani, S., Demner-Fushman, D., Thoma, G.R.: Image retrieval from scientific publications: Text and image content processing to separate multipanel figures. *J. Assoc. Inf. Sci. Technol.* **64**(5), 893–908 (2013). DOI 10.1002/asi.22810
2. Bengio, Y.: Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2**(1), 1–127 (2009). DOI 10.1561/22000000006
3. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **35**(8), 1798–1828 (2013). DOI 10.1109/TPAMI.2013.50
4. Bishop, C.M.: *Pattern Recognition and Machine Learning*, chap. 1.5 (Decision Theory), pp. 38–47. Springer, Secaucus, NJ, USA (2006)
5. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, pp. 161–168. ACM, New York, NY, USA (2006). DOI 10.1145/1143844.1143865
6. Chatzichristofis, S.A., Boutalis, Y.S.: CEDD: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: *Computer Vision Systems, LNCS*, vol. 5008, pp. 312–322. Springer (2008). DOI 10.1007/978-3-540-79547-6_30
7. Chen, N., Blostein, D.: A survey of document image classification: Problem statement, classifier architecture and performance evaluation. *Int. J. Doc. Anal. Recognit.* **10**(1), 1–16 (2007). DOI 10.1007/s10032-006-0020-2
8. Chhatkuli, A., Foncubierta-Rodríguez, A., Markonis, D., Meriaudeau, F., Müller, H.: Separating compound figures in journal articles to allow for subfigure classification. *Proc. SPIE* **8674**, 86,740J–12 (2013). DOI 10.1117/12.2007897
9. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*, 2nd edn., chap. 7 (Model Assessment and Selection), pp. 219–260. Springer, New York (2009)
10. García Seco de Herrera, A., Kalpathy-Cramer, J., Demner-Fushman, D., Antani, S., Müller, H.: Overview of the ImageCLEF 2013 medical tasks. In: *CLEF 2013 Working Notes, CEUR Proc.*, vol. 1179 (2013). URL <http://ceur-ws.org/Vol-1179/>
11. García Seco de Herrera, A., Müller, H., Bromuri, S.: Overview of the ImageCLEF 2015 medical classification task. In: *CLEF 2015 Working Notes, CEUR Proc.*, vol. 1391 (2015). URL <http://ceur-ws.org/Vol-1391/>
12. Kalpathy-Cramer, J., de Herrera, A.G.S., Demner-Fushman, D., Antani, S., Bedrick, S., Müller, H.: Evaluating performance of biomedical image retrieval systems—an overview of the medical image retrieval task at ImageCLEF 2004–2013. *Computerized Medical Imaging and Graphics* **39**(0), 55–61 (2015). DOI 10.1016/j.compmedimag.2014.03.004. Medical visual information analysis and retrieval
13. Kitanovski, I., Dimitrovski, I., Loskovska, S.: FCSE at medical tasks of ImageCLEF 2013. In: *CLEF 2013 Working Notes, CEUR Proc.*, vol. 1179 (2013). URL <http://ceur-ws.org/Vol-1179/>
14. Kou, G., Lu, Y., Peng, Y., Shi, Y.: Evaluation of classification algorithms using MCDM and rank correlation. *International Journal of Information Technology & Decision Making* **11**(01), 197–225 (2012). DOI 10.1142/S0219622012500095

15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: F. Pereira, C. Burges, L. Bottou, K. Weinberger (eds.) *Advances in Neural Information Processing Systems* 25, pp. 1097–1105. Curran Associates, Inc. (2012)
16. Mitchell, T.M.: *Machine Learning*, chap. 5 (Evaluating Hypotheses), pp. 128–153. McGraw-Hill, New York (1997)
17. Murphy, R.F., Velliste, M., Yao, J., Porreca, G.: Searching online journals for fluorescence microscope images depicting protein subcellular location patterns. In: *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering, BIBE '01*, pp. 119–128. IEEE Computer Society, Washington, DC, USA (2001)
18. Pelka, O., Friedrich, C.M.: FHDO Biomedical Computer Science Group at medical classification task of ImageCLEF 2015. In: *CLEF 2015 Working Notes, CEUR Workshop Proceedings, ISSN 1613-0073*, vol. 1391 (2015). URL <http://ceur-ws.org/Vol-1391/14-CR.pdf>
19. Qian, Y., Murphy, R.F.: Improved recognition of figures containing fluorescence microscope images in online journal articles using graphical models. *Bioinformatics* **24**(4), 569–576 (2008). DOI 10.1093/bioinformatics/btm561
20. Santosh, K., Xue, Z., Antani, S., Thoma, G.: NLM at ImageCLEF 2015: Biomedical multipanel figure separation. In: *CLEF 2015 Working Notes, CEUR Proc.*, vol. 1391 (2015). URL <http://ceur-ws.org/Vol-1391/>
21. Shatkay, H., Chen, N., Blostein, D.: Integrating image data into biomedical text categorization. *Bioinformatics* **22**(14), e446–e453 (2006). DOI 10.1093/bioinformatics/btl235
22. Simpson, M.S., You, D., Rahman, M.M., Xue, Z., Demner-Fushman, D., Antani, S., Thoma, G.: Literature-based biomedical image classification and retrieval. *Comput. Med. Imag. Graph.* **39**, 3–13 (2015). DOI <http://dx.doi.org/10.1016/j.compmedimag.2014.06.006>
23. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1470–1477. IEEE (2003)
24. Smith-Miles, K.A.: Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.* **41**(1), 6:1–6:25 (2009). DOI 10.1145/1456650.1456656
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
26. Taschwer, M., Marques, O.: AAUITEC at ImageCLEF 2015: Compound figure separation. In: *CLEF 2015 Working Notes, CEUR Proc.*, vol. 1391 (2015). URL <http://ceur-ws.org/Vol-1391/>
27. Taschwer, M., Marques, O.: Compound figure separation combining edge and band separator detection. In: Q. Tian, N. Sebe, G.J. Qi, B. Huet, R. Hong, X. Liu (eds.) *MultiMedia Modeling, Lecture Notes in Computer Science*, vol. 9516, pp. 162–173. Springer International Publishing (2016). DOI 10.1007/978-3-319-27671-7_14
28. Wang, X., Jiang, X., Kolagunda, A., Shatkay, H., Kambhamettu, C.: CIS UDEL working notes on ImageCLEF 2015: Compound figure detection task. In: *CLEF 2015 Working Notes, CEUR Workshop Proceedings, ISSN 1613-0073*, vol. 1391 (2015). URL <http://ceur-ws.org/Vol-1391/65-CR.pdf>
29. Yuan, X., Ang, D.: A novel figure panel classification and extraction method for document image understanding. *Int. J. Data Min. Bioinformatics* **9**(1), 22–36 (2014). DOI 10.1504/IJDMB.2014.057779